Design and Characteristics of a Split Hopkinson Pressure Bar Apparatus

Radek Glaser, University of Maine - Main
Jesse Haines, University of Maine
Christopher Knight, University of Maine
DESIGN AND CHARACTERISTICS OF A SPLIT HOPKINSON PRESSURE BAR APPARATUS

Jesse Haines
Christopher Knight
Radek Glaser

May 10th, 2007
University Of Maine
Mechanical Engineering
5711 Boardman Hall
Orono, ME 04469

ABSTRACT

A Split Hopkinson Bar Apparatus, also known as Kolsky Bar that is capable of conducting compressive strain rate testing in the approximate ranges from 50 to 10^4 in. /in. /sec. was designed as a part of a Senior Design Project. Generally, this device is similar to that first used by Kolsky in 1949. The design of this device is presented here in two stages:
1. Research, design and manufacturing of the Stress Generating System
2. Experimental Part – Testing of the apparatus to obtain necessary data.

The present phase of the design was focused mostly on the stress generating system for the apparatus. Final experiments were conducted to obtain the calibration curve and the shapes of generated waves. Not only the description of the design process is presented, but also the final details including the assembly and testing are given. Also some valuable suggestions for future work that is necessary to the make the apparatus better is given.

ACKNOWLEDGMENTS

We would like to express our thanks for the assistance and help we received from Michael Knight, our advisor Prof. Vincent Caccese, Arthur Pete, Prof. Michael Peterson and the Staff of the Advanced Manufacturing Center during the lifetime of the Split Hopkinson Pressure Bar Project.
# Table of Contents

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>1</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>Objectives of the Project</td>
<td>3</td>
</tr>
<tr>
<td>Researching the Split Hopkinson Bar Apparatus</td>
<td>3</td>
</tr>
<tr>
<td>Design Approach</td>
<td>4</td>
</tr>
<tr>
<td>Anticipated results</td>
<td>4</td>
</tr>
<tr>
<td>Final Design</td>
<td>5</td>
</tr>
<tr>
<td>Design Overview</td>
<td>5</td>
</tr>
<tr>
<td>Split Hopkinson Bar Parts</td>
<td>7</td>
</tr>
<tr>
<td>Control System</td>
<td>9</td>
</tr>
<tr>
<td>Design Testing-Experimental</td>
<td>12</td>
</tr>
<tr>
<td>Results and Conclusion</td>
<td>15</td>
</tr>
<tr>
<td>Recommendations</td>
<td>15</td>
</tr>
<tr>
<td>Appendix 1-Abandoned ideas</td>
<td>16</td>
</tr>
<tr>
<td>Appendix 2-Additional graphs of the waves</td>
<td>18</td>
</tr>
<tr>
<td>Appendix 3- Analog and Digital Pin IN/OUT connectors</td>
<td>20</td>
</tr>
<tr>
<td>Appendix 4-Program Codes</td>
<td>22</td>
</tr>
<tr>
<td>Endnotes</td>
<td>22</td>
</tr>
</tbody>
</table>
INTRODUCTION

The Split Hopkinson Pressure Bar design began during the Fall semester of 2006 in the Mechanical Engineering Senior Design class at the University Of Maine. Our group that consisted of Chris Knight, Jesse Haines, and Radek Glaser chose the project based not only on its high potential and long time use, but also because of its multiple capabilities. The Split Hopkinson Bar has the ability to test the compressive strength of variety materials, such as: laminated composites, ceramics, different metals and ballistics. The apparatus is being designed with longevity and flexibility of the operational characteristics in mind. This will allow for a wide range of testing and limit future costs associated with the project. Evidently, Split Hopkinson Pressure Bar Apparatus Project has the potential for future growth through an external or departmental funding associated with the testing.

OBJECTIVES OF THE PROJECT

During the project the main focus was placed on the design, assembly and testing of the stress generating system of the Split Hopkinson Pressure Bar Apparatus. The following were the main objectives of the design process of the apparatus:

Design and build a fully functional apparatus that would be capable of:

* Operating in a safe manner
* Producing striker velocities in the range of 100 -160 ft/sec.
* Producing pressure-velocity calibration curve
* Generating impact-compression tests at strain rates ranging from 50 to 10^4 in./in./sec.
* Generating propagation waves that can be used to determine strain-stress relationship

RESEARCHING THE SPLIT HOPKINSON BAR APPARATUS

Early in the design class during the initial stages of our Split Hopkinson Bar Apparatus Design we attempted to find as much useful information as possible about the apparatus. Our main task was to find the most reliable information that could lead us to successful completion of our project including all our objectives. The main concern was to find an efficient way to allow for a sudden flow of the compressed gas from the tank to the
launching tube so the striker could be accelerated properly. After researching not only many websites and publications on different versions of the Split Hopkinson Bars around the world, we discovered that in many cases this crucial information was not really available for public access. Finally, we found an unclassified army materials and mechanic research booklet titled “Design and Operating Characteristics of a Split Hopkinson Pressure Bar Apparatus”.\(^1\) This booklet is where much of our own research and design information of our Split Hopkinson Pressure Bar Apparatus came from. From the research Phase we moved to design approach part that is included in the next section.

**DESIGN APPROACH**

As stated before, most of the details in the design of our Split Hopkinson Bar Apparatus came from the military booklet due to very meticulous information included in it. The booklet goes into large detail of the importance of the gun and its ability to perform very quick, large impulse, strain test. Although we tried to keep the details of the military design as best we could, soon we found out that we did not have the manufacturing capabilities or necessary budget. So we came with our own design that scaled down and simplified the original apparatus. While designing the apparatus we kept the important parameters and characteristics of the design the same, to have equal or even better results. While the military design had 100 inch long launching tube we used 24 inch tube due to manufacturing issues. The original design used a solenoid valve that operated up to 300psi, but we increased operational pressure up to 750 psi by using appropriate solenoid valve\(^2\) Also the frictional aspect was partially changed. Thanks to an explanation about potential problems with lubrication during the operation of the apparatus we were able to avoid the reduction of velocity that was due to the excess of lubrication. Being able to refer to the military design, not only made it much easier and faster for us to design our Split Hopkinson Pressure Bar Apparatus, but also helped us to avoid potentially unfavorable situations during the design process. Certain analytical procedures were conducted to get numerical results for the final design. These anticipated results follow in the next section.

**ANTICIPATED RESULTS**

In most of the designs we can expect that our results will fall into a certain range. In our case it came from computational model based on simple Newtonian dynamics
relationships with many details omitted to keep our model simple. The following are the steps that lead to final design.

**Equations of Motion for Striker Bar**

![Fig1. Simple FBD](image)

\[ a_x = \frac{PA}{m}, \quad x = \frac{1}{2} \frac{PA}{m} t^2, \quad v_x = \frac{PA}{m} t \]

\[ t = \sqrt{\frac{2x}{\frac{PA}{m}}}, \quad v_x = \sqrt{\frac{2PA}{m} L} \]

Where the individual terms are:

- \( v_x \) - velocity of the projectile,
- \( P \) - initial pressure of compressed air tank,
- \( A \) - area of the projectile,
- \( L \) - travel distance (barrel length)
- \( m \) - mass of the projectile
- \( x \) - displacement of the projectile.
- \( t \) - average travel time
- \( a \) - acceleration

The values of \( A \) and \( m \) depend upon the radius and density \( \rho \) of the projectile itself.

Force Balance in the \( x \) direction, neglecting kinetic and air friction gives the following:

\[ \sum F_x = PA = ma_{x-direction} \]

From the relationships above we obtained anticipated velocities and travel times for our initial model. It is shown on the following graph.

![Fig2. Shows velocity profile (blue) and time of travel (pink).](image)

These were anticipated results and the sections below will show how much the final results differ from the optimal.

**FINAL DESIGN**

**DESIGN OVERVIEW**

The Split Hopkinson Bar Apparatus works with compressed gas. The compressed nitrogen tank has an initial pressure of 2200 psig. There is a high pressure regulator that shows the current pressure in the tank and also limits the delivery pressure to the solenoid valve. The pressure of the exiting nitrogen can be set in variety of increments up to a maximum of 750 psi, which is the safe working pressure of the solenoid valve. Our computerized control system opens the
solenoid valve letting the gas expand to the launching tube pushing the striker out. When the striker reaches the end of the barrel it blocks the laser beam that is shining at the sensor, changing the voltage seen by the control system. At the same time striker bar impacts the incident bar producing the propagation wave through the bars. The control system closes the valve blocking off the delivery of nitrogen. The velocity of the striker is measured from the time the solenoid valve is opened to the time the voltage changes closing the solenoid valve. The excess of the pressure from the launching tube is released by a hand operated valve and then the vacuum suction is applied to move the striker bar to starting position.

Then the wave travels through the incident bar, sample and to the output bar. Once this wave reaches the test specimen, part of it is transmitted through the specimen and part of it reflects back through the incident bar.

Using one dimensional wave propagation analysis we can determine high strain rate stress-strain curves from the measurements of strain in the incident and output bars. In order to obtain correct wave propagation the striker bar must reach necessary velocity before it hits the incident bar. The wave is measured through a set of strain gages located on the incident and transmitter bar. The test specimen is located between the incident bar and

---

**Fig.3** shows a diagram of the apparatus.

The second part of the apparatus is the strain measuring section. It is composed of a striker bar, an incident bar, the test specimen and the output bar. The impact of the striker bar on incident bar produces a rectangular compression wave of well-defined amplitude and length in the incident bar.
transmitter bar. Depending on the type of specimen and the impacting force we would get a part of the wave is transmitted back through the incident bar, and the rest is sent through the test specimen to the transmitter bar. Waves can be seen on the oscilloscope that is attached to the measuring system.

**SPLIT HOPKINSON BAR PARTS**

**Launching Tube**

A crucial component of the stress generating system is the launching tube. Table below shows two different barrels used for testing.

<table>
<thead>
<tr>
<th>TUBE</th>
<th>O. D. (in)</th>
<th>I. D. (in)</th>
<th>Length(in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0 to .77</td>
<td>.61</td>
<td>33.0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.731</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 1 Shows Tube Options

The barrel is screwed onto a flange connecting it to the rest of the apparatus. Within the flange at the end of the barrel there is an inside taper that prevents the striker bar from over travel while being retracted by the vacuum pump. Therefore the geometry of the inside of the flange is changed resembling a converging nozzle. This definitely reduces the flow through the apparatus and pressure exerted on the striker bar. Figure below shows barrel, flange and solenoid valve currently being used during testing.

**Striker Bars**

Both striker bars were made of 4130 steel and have an outside diameter corresponding to the inside diameter of the launching tubes. The length of the striker bar that is currently being used is 3.125 inch, but this can vary depending on the desired propagation wavelength and the tube used for testing. Appropriate lubricant is be used sparingly inside the barrel and on the striker to reduce friction between these two parts during the discharge procedure.
**Solenoid Valve, Gas Tank and Regulator**

The most important component of the apparatus is the quick opening solenoid valve that releases compressed nitrogen which forces the striker down the barrel. The solenoid valve is an ASCO 8223G0053 that is rated for up to 750 psi. The compressed nitrogen comes from a nitrogen tank with an initial pressure of 2200 psi. The tank is connected to the solenoid valve through sealed piping and a pressure regulator. The pressure regulator allows us to deliver required flow by regulating the exiting delivery pressure up to a limiting pressure of 900 psi. The solenoid valve releases the gas to the launching tube and ultimately the striker exits with desired pressure and velocity. The maximum operating pressure for the valve is 750 psi.

**Vacuum Pump**

The vacuum pump is used to return the striker bar to the starting position at the rear end of the barrel. The 0.4 W pump is connected to the flange that is connected to the barrel and solenoid. As soon as the solenoid valve shuts off the delivery of nitrogen to the launching tube and the excess gases are manually released from the tube, the vacuum pump is activated by the computer and creates vacuum in the barrel resulting in the bullet to be retracted to the starting position.

**Incident, Transmitter and Momentum Bars**

The strain measurement system is made up of three bars and the test specimen. The incident bar is the bar that the striker comes in contact with when shot. The test specimen is held between the incident bar and the transmitter bar. At the end of the assembly there is a momentum bar that absorbs the remaining energy from the
striker bar. All bars are made of steel and are 18.25 inches long with a diameter of .5 inch.

**Stand**

We decided to use a 10-feet long, 4x4 inch I-beam as a reference plane that most the apparatus sits on. Only gas tank and pressure regulator are not located on the I-beam. The apparatus takes up roughly 3 feet of the beam. The rest of the beam is used to support the setup that the striker bar hits. Five legs made from hollow two by four inch square tubing found in Crosby Lab support the beam.

**CONTROL SYSTEM**

**Control Box**

The control box houses the electronics needed to trigger and close the solenoid valve. It includes relays, wires, fuse, inlets and outlets for signal receiving processing and delivery to and from computer vacuum pump and Photo-sensor. Data in Appendix 3

**Photo Sensor**

The Photo sensor utilizes a potentiometer in series with a photo resistor, which in operation varies the voltage across the Photo sensor depending upon the intensity of light on its surface, which can also be tuned using the potentiometer. This voltage is high when light is incident on its surface and low when the sensor is blocked by a projectile. Changes in voltage are processed and

---

Fig.9 Incident and transmitter bar

Fig.10 Stand assembly.

Fig.11 Control Box.
delivered to the computer that shuts off the solenoid valve and cutting off the nitrogen supply.

**Laser**

We used small commercially available hunting laser with a power level of roughly 5mW and is powered by 3 small 1.5 volts batteries or by an external DC voltage supply. Below are two pictures of the laser and photo sensor that show how the striker comes out and blocks the laser, which changes the voltage read by the control system and turns the solenoid valve off.

![Fig.12 On the left laser active. On the right striker bar exiting the tube and disrupting the laser light](image)

It ties all other systems together in order to fire the gas gun.

![Fig.13 Daqbook](image)

As mentioned earlier Daqbook is connected to the computer and the control box allowing for signal processing

**Daqbook**

A Daqbook Iotech Daqbook 100, is used as the focal point of our control system. The daqbook is controlled by a Delphi programs which access several features such an Intel 9513 Timer Chip, Digital I/O ports for controlling relays and several analog ports for taking voltage readings.

![Fig.14 Computer connected to the Daqbook](image)
Control Program

The control program is written Delphi 5 and is derived from software used by Professor Vincent Caccese for other applications. The program uses Windows dlls to control Daqbook functions which can then be accessed by our control program written in Delphi. It contains functions which read voltages and turn on/off relays. Using these features in combination makes it possible to control the Split Hopkinson Bar Apparatus. Programs are listed in Appendix 4.

Control Program Main Form

The following figure is a picture of the main screen to control Split Hopkinson Bar Apparatus. In the picture you can see where the solenoid valve and vacuum pump are turned on and off, the voltage readout for the laser at the end of the barrel, and the all important button which fires the striker.

![Control Program Main Form](image)

Fig. 15 Control Program Main Form
Velocity Calibration

First test performed with the Split Hopkinson Bar Apparatus was velocity calibration. Pressure controller was used to incrementally release a certain amount of gas to the solenoid valve. The test was performed using pressures ranging from 25 psi to 550 psi. The calibration test was done to see what velocities would be reached for corresponding various shooting pressures. We performed this procedure several times to obtain the repeatability of results. The program described in the previous section allowed us to read the results: velocity and travel time off the computer screen thanks to the interface. We obtained three sets of values during our experiments. Later on those values were used to produce calibration curve using Microsoft Excel. Values obtained during those three experiments are listed in the table that follows.

<table>
<thead>
<tr>
<th>Test Date</th>
<th>Pressure (psig)</th>
<th>Travel Time (sec)</th>
<th>Velocity (ft/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-19-2007</td>
<td>50</td>
<td>0.094</td>
<td>21.2766</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>0.095</td>
<td>21.05263</td>
</tr>
<tr>
<td></td>
<td>165</td>
<td>0.07</td>
<td>28.57143</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>0.07</td>
<td>28.57143</td>
</tr>
<tr>
<td></td>
<td>360</td>
<td>0.059</td>
<td>33.89831</td>
</tr>
<tr>
<td></td>
<td>450</td>
<td>0.06</td>
<td>33.33333</td>
</tr>
<tr>
<td></td>
<td>550</td>
<td>0.062</td>
<td>32.25806</td>
</tr>
<tr>
<td>04-30-2007</td>
<td>25</td>
<td>0.131</td>
<td>15.26718</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.092</td>
<td>21.73913</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.082</td>
<td>24.39024</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.074</td>
<td>27.02703</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.074</td>
<td>27.02703</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.064</td>
<td>31.25</td>
</tr>
<tr>
<td>05-01-2007</td>
<td>75</td>
<td>0.089</td>
<td>22.47191</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>0.081</td>
<td>24.69136</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>0.09</td>
<td>22.22222</td>
</tr>
<tr>
<td></td>
<td>140</td>
<td>0.077</td>
<td>25.97403</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.074</td>
<td>27.02703</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>0.071</td>
<td>28.16901</td>
</tr>
<tr>
<td></td>
<td>220</td>
<td>0.07</td>
<td>28.57143</td>
</tr>
<tr>
<td></td>
<td>260</td>
<td>0.06</td>
<td>33.33333</td>
</tr>
</tbody>
</table>

Using the values from table above, a velocity versus pressure graph was made that is located in the figure on the following page. On the chart the best fitted line is\[ y = 7.9507x^{0.2387} \] This line gives the best estimate of what the striker velocity will be for a given pressure.
Strain Measurements

The strain-stress measurement was performed at the same time as the velocity test. Strain is measured in the incident and first transmitter bars. Three strain gages were strategically located on the bars, two on the incident bar, and one on the transmitter bar. The strain gages measured the wave that propagated through the bars. When the striker hits the incident bar, the wave travels through the bar until it reaches the test specimen. Depending on the specimen, some of the wave reflects back through the incident bar while the rest travels through the specimen and through the transmitter bar. For the strain test, the strain gages were connected to a P-3500 strain amplifier and the output was shown on a Hewlett Packard Oscilloscope. The following two figures located on the next page show the expected waves traveling through the incident and transmitter bars. The figures following are the images of propagation waves gathered during our experiment. They are given as copy of the printout and also pictures taken from the oscilloscope screen.
After problems arose with the strain gages on the incident bar, only the wave from the strain gage on the transmitter bar could be seen. The rest of the figures are listed in Appendix 2.
**Results and Conclusion**

Overall the designing and building of the Split Hopkinson Pressure Bar Apparatus was a success. We went from the thought of just designing the gun at the beginning of the year to full building and testing at the end of last month. Although we were not able to obtain the predicted velocities we reached striker velocities in the range of over 35 ft/sec while not even shooting at full potential. During the same velocity test we were able to gather several strain curves from an oscilloscope, which compared fairly well to those found in the military booklet and others online. Possible reasons for reduced velocities:

1. Design calculations were based on the flow in a straight pipe - not through converging nozzle, but reduced throat area was necessary to keep the striker bar in the right position.
2. Design calculations did not include air frictional factors-back pressure.
3. Friction of the striker bar against the barrel was not included in estimation.
4. Experimental velocities were reduced by the uncertainty connected to computer hardware. Because a minimum time increment of the counter was 15 milliseconds at high pressures did not give enough room to measure elapsed time. Evidently, this design class and project have given us the best learning experience. Undoubtedly we will encounter similarly challenging tasks in the real engineering world.

**Recommendations**

1. To allow for the diversification of testing out technical advisor Professor Vince Caccese suggested putting a steel plate at the end of the barrel. It will allow holding an object that would undergo destructive testing.

2. More velocity calibration test performed at larger pressures is required. At the same time find a way to get the time increments smaller in the computer program for better velocity accuracy.

3. Search for better strain gages. Testing at high pressures and velocities broke the gages located on the incident bar; therefore more durable ones should be used in the future.

4. It will absolutely essential to resolve the counter problem, so the system can use smaller time increments to do the measurements.
APPENDIX 1

The following are the transitional ideas-options for the various elements of the apparatus that were put aside during the process and reasons for these decisions.

Piston
Originally the group considered the idea of using a piston with a striker bar part connected to it to allow the piston much higher accelerations and a much smaller barrel length. This idea was scrapped in favor of a longer barrel.

Fabricated Pressure Vessel
Early in the project it was decided that the air supply for the air gun would be a pressure vessel made of a hollow tube and side plates. Initial dimensions of 8.5 inches in diameter and 11 inches long were chosen for initial performance calculations, but the idea was scrapped in favor of prefabricated compressed gas tanks for safety and ease of use.

Maximum Optimal Sized Barrel
The original plan for the group was to use the general dimensions of most other split hopkinson bar projects and build something around the same dimensions we found in other sources. Our dimensions were later scaled down due to in ability to produce or subcontract the manufacturing of a very long barrel with a very tight tolerances and within a reasonable budget limits. At the beginning of the semester, we looked into buying a barrel that met the requirements of the military specifications. The size of the barrel was 100 inches long with a 1-inch inner diameter and 3 inch outer diameter. The 1-inch thickness was to meet the straightness requirement of the barrel and not a safety factor. After talking several machine shops that could not do the drilling until late 2007, or who gave quotes exceeding our budget we decided to downsize. The barrel we have chosen was an original part of a shoot gun drilled gun barrel, which was 30 inches long, has an inner diameter of 0.71 inches. Although the cost of this barrel was very low comparing to the previous options, final on site inspection of the tube excluded it from a potential application. It turned out that the inside diameter was tapered (0.712 to 0.725), therefore making it unable to have the desired tolerances.

Mylar Discs
Initially we were considering using mylar membranes/disks that are melted by running
a current through a small wire in contact with it to release the pressure stored in their air tank. The idea was abandoned due to the possibility of clogging of the launch tube by the broken disks.

**Iris Valves**

Considering affordable iris valves we discovered that iris valves are made to handle gravity fed systems and would not have stood up to the high pressures that it would have been subjected to in our apparatus.

**Rupture Disks**

Another option for a fast release of gas from the reservoir to the launch tube was an aluminium rupture disk. Rupture disks would have been an excellent method seeing as there is practically no opening time as with solenoid valves, but the cost per disk and high uncertainty with burst pressure eliminated it.

**Backup Gas Release**

It was suggested midway through the project that our group should into a backup system for gas release in case of a solenoid valve failure. This wasn’t entirely a dead end or a scrapped idea per say, but it just wasn’t something we could afford to invest the time in. It may however be an excellent idea for next semester were time permitting.
APPENDIX 2

More Strain Measurement Graphs

Fig 23: Wave in Incident Bar with Composite Test Specimen

Fig 24: Wave Through Transmitter Bar shot at 260 psi
Fig 25: Wave Through Transmitter Bar shot at 220 psi

Fig 26: Wave Through Transmitter Bar shot at 180 psi

Fig 27: Wave Through Transmitter Bar shot at 150 psi
APPENDIX 3

Shows connections of the Analog and Digital Ports in the control Box. Please refer to the DAQBOOK 00 Users manual for further reference.

1. ANALOG PORT (front View)

2. DIGITAL PORT (front View)
3. OUR CONNECTIONS (Back View-from the inside of the Control Box)

<table>
<thead>
<tr>
<th>PIN CONNECTOR NUMBER</th>
<th>ANALOG I/O</th>
<th>DIGITAL I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>fan voltage (digital ground)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Voltage to sensor BLACK, dg</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>+ 5 V supply to fan</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Signal from sensor-RED</td>
<td>Vacuum relay digital ground</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>Solenoid relay digital ground</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>+ from thermocouple</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Ground for Strain Gage Input</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Signal from sensor-GREEN</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>- from thermocouple</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Strain Gage input from amp</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Strain Gage input from amp</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Strain Gage input from amp</td>
<td>Digital out to sensor-WHITE</td>
</tr>
<tr>
<td>36</td>
<td>Strain Gage input from amp</td>
<td>Dig. Out to Solenoid Relay</td>
</tr>
<tr>
<td>37</td>
<td>Strain Gage input from amp</td>
<td>Dig. Out to Vacuum Pump Rel</td>
</tr>
</tbody>
</table>
APPENDIX 4

Includes the computer code used to control Split Hopkinson Pressure Bar Apparatus. The `GasGunC` file defines constants, the `GasGunMain` file initializes the hardware and defines subroutines, the `GasGunInterface` file is the code which the user controls the daq with, the `daqx` file contains the commands that Delphi uses to control the daqbook, and the `Errex` file is used to give error commands from the daqbook to the user. The picture is of the form associated with the interface file that the user interacts with to run subroutines within the interface file.

ENDNOTES


2 [www.asco.com](http://www.asco.com)

3 [www.asco.com](http://www.asco.com)

unit GasGunC;

interface

uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, DaqX;

const

DIOPort_A = 0;
DIOPort_B = 1;
DIOPort_C = 2;

DIO_Write = false;
DIO_Read = true;

VacuumPumpRelay = 0;
SolenoidValveRelay = 1;
SensorVoltageSource = 2;
SensorChan = 7;
Gauge1Chan = 0;
Gauge2Chan = 1;
Gauge3Chan = 2;

var

DaqHandle00 : DaqHandleT;
daqDeviceType00 : DaqDacDeviceType;
daqIODeviceType00 : DaqIODeviceType;
DIOConfig : byte;
DIODevPort : DWORD;
DIODevType : DaqIODeviceType;
DIOExpansionPort : DaqIOExpansionPort;
DIOWhichDevice : DWord;

implementation

end.
unit GasGunMain;

interface

uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls, TeeProcs, TeEngine, Chart, GasGunInterface, Errex, GasGunC, AppEvnts, daqx, Series;

type
TGasGunMainForm = class(TForm)
RelatStatuGB: TGroupBox;
vrs: TEdit;
Label1: TLabel;
Label2: TLabel;
srs: TEdit;
SensorStatusGB: TGroupBox;
StrainGaugeStatusGB: TGroupBox;
Label4: TLabel;
Label6: TLabel;
Label7: TLabel;
StrainGauge1VoltageEB: TEdit;
StrainGauge2VoltageEB: TEdit;
StrainGauge3VoltageEB: TEdit;
RoomtempGB: TGroupBox;
RoomAirTempEB: TEdit;
Label8: TLabel;
GroupBox1: TGroupBox;
FireControlBT: TButton;
SensorVoltageSourceGB: TGroupBox;
Label9: TLabel;
SensorVoltageEB: TEdit;
VoltageOnBT: TButton;
VoltageOffBT: TButton;
VacPumpOn: TButton;
VacPumpOff: TButton;
SolValveOff: TButton;
SolValveOn: TButton;
Label10: TLabel;
Label11: TLabel;
RadioButton1: TRadioButton;
Label14: TLabel;
GroupBox2: TGroupBox;
Edit1: TEdit;
Label16: TLabel;
RadioButton3: TRadioButton;
RadioButton4: TRadioButton;
Label15: TLabel;
Timer2: TTimer;
GroupBox3: TGroupBox;
Chart1: TChart;
Series1: TPointSeries;
Series2: TPointSeries;
Series3: TPointSeries;
procedure FormCreate(Sender: TObject);
procedure VacPumpOnClick(Sender: TObject);
procedure VacPumpOffClick(Sender: TObject);
procedure SolValveonClick(Sender: TObject);
procedure SolValveoffClick(Sender: TObject);
procedure VoltageOnBTClick(Sender: TObject);
procedure VoltageOffBTClick(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure FireControlBTClick(Sender: TObject);

private
{ Private declarations }

public
{ Public declarations }

end;

var
GasGunMainForm: TGasGunMainForm;
procedure TGasGunMainForm.FormCreate(Sender: TObject);
begin
  FireTheDaq(1);  
  daq9513SetMasterMode(DaqHandle00, DiodtLocal9513, 0, 1, DcsF4, false, false, DtodDisabled);  
  daq9513SetCtrMode(DaqHandle00, DiodtLocal9513, 0, 1, DgcNoGating, true, DcsF4, false, false, false, false, true, DocInactiveLow);  
  GasGunInterface_Frm.InitRelays;  
  {Turning all relay status settings to off}
  vrs.text := 'OFF';
  srs.text := 'OFF';
  Counter := 0;
end;

{Turning on Vacuum Pump Relay via button}
procedure TGasGunMainForm.VacPumpOnClick(Sender: TObject);
begin
  GasGunInterface_Frm.TurnRelayOn(VacuumPumpRelay);
  vrs.text := 'ON';
end;

{Turning off Vacuum Pump Relay via button}
procedure TGasGunMainForm.VacPumpOffClick(Sender: TObject);
begin
  GasGunInterface_Frm.TurnRelayOff(VacuumPumpRelay);
  vrs.text := 'OFF';
end;

{Turning on Solenoid Valve Relay via button}
procedure TGasGunMainForm.SolValveOnClick(Sender: TObject);
begin
  GasGunInterface_Frm.TurnRelayOn(SolenoidValveRelay);
  srs.text := 'ON';
end;

{Turning off Solenoid Valve Relay via button}
procedure TGasGunMainForm.SolValveOffClick(Sender: TObject);
begin
  GasGunInterface_Frm.TurnRelayOff(SolenoidValveRelay);
  srs.text := 'OFF';
end;

procedure TGasGunMainForm.VoltageOnBTClick(Sender: TObject);
begin
  GasGunInterface_Frm.TurnRelayOn(SensorVoltageSource);
end;

procedure TGasGunMainForm.VoltageOffBTClick(Sender: TObject);
begin
  GasGunInterface_Frm.TurnRelayOff(SensorVoltageSource);
end;

**********************************************************************
*       Firing Controls                                                *
**********************************************************************
begin
  SensorVoltage := Scandata.ReadOneChannel(SensorChan);
  SensorVoltageEB.Text := FloatToStr(SensorVoltage);
  StrainGauge1Voltage := Scandata.ReadOneChannel(Gauge1Chan);
  StrainGauge1VoltageEB.Text := FloatToStr(StrainGauge1Voltage);
  StrainGauge2Voltage := Scandata.ReadOneChannel(Gauge2Chan);
  StrainGauge2VoltageEB.Text := FloatToStr(StrainGauge2Voltage);
  StrainGauge3Voltage := Scandata.ReadOneChannel(Gauge3Chan);
  StrainGauge3VoltageEB.Text := FloatToStr(StrainGauge3Voltage);
end;

procedure TGasGunMainForm.FireControlBTClick(Sender: TObject);
var
  loop : boolean;
  SensorV : real;
  str1 : string;
  str2 : string;
  str3 : string;
  DateTime : TDateTime;
  Hour, Min1, Min2, Sec1, MSec1, Sec2, MSec2: Word;
  eTime : real;
  counts : WORD;
  count1 : WORD;
  count2 : WORD;
  count3 : WORD;
  something : WORD;
  velocity: real;
  inc : word;
begin
  if Radiobutton1.Checked then
  begin
    loop := true;
    count1 := 0;
    count2 := 0;
    count3 := 0;
    series1.Clear;
    series2.Clear;
    series3.Clear;
    //DecodeTime(Time,Hour,Min1,Sec1,MSec1); //System Based Time Measurement
    GasGunInterface_Frm.TurnRelayOn(SolenoidValveRelay);
    daq9513SetHold(DaqHandle00, DiodtLocal9513,0, 1, 0);
    daq9513MultCtrl(DaqHandle00, DiodtLocal9513, 0, DmccArm, true, false, false, false);
    daq9513MultCtrl(DaqHandle00, DiodtLocal9513, 0, DmccSave, true, false, false, false);
    daq9513GetHold(DaqHandle00, DiodtLocal9513, 0, 1, count1);
    while loop do
    begin
      daq9513MultCtrl(DaqHandle00, DiodtLocal9513, 0, DmccSave, true, false, false, false);
      daq9513GetHold(DaqHandle00, DiodtLocal9513, 0, 1, count2);
      SensorV := Scandata.ReadOneChannel(SensorChan);
      if SensorV > 3.5 then Loop := false;
    end;
    daq9513MultCtrl(DaqHandle00, DiodtLocal9513, 0, DmccDisarm, true, false, false, false);
    GasGunInterface_Frm.TurnRelayOff(SolenoidValveRelay);
{DecodeTime(Time,Hour,Min2,Sec2,MSec2); //System Time Based measurement

eTime := (Sec2-Sec1) + (MSec2-Msec1)/1000;} //System Time Based measurement

If (count2-count1) > 0 Then
  begin
  str1 := 'Projectile Velocity ' + FloatToStr(2/((count2 - count1)/1000)) + ' ft/s';
  str2 := 'Projectile Travel Time ' + FloatToStr((count2-count1)/1000) + 's';
  showmessage(str1);
  showmessage(str2);
  end
Else
  If (count2-count1) = 0 Then
    Begin
    str1 := 'Projectile Velocity - ' + 'infinity' + ' ft/s';
    str2 := 'Projectile Travel Time ' + '0' + 's';
    showmessage(str1);
    showmessage(str2);
    radiobutton1.Checked := False;
    End;
  end;
End;

****************************************************************************
*                      Strain Gauge Measurements
***************************************************************************

loop := true;
something := 1;
daq9513SetHold(DaqHandle00, DiodtLocal9513,0, 1, 0);
daq9513MultCtrl(DaqHandle00, DiodtLocal9513, 0, DmccArm, true, false, false, false, false);
daq9513MultCtrl(DaqHandle00, DiodtLocal9513, 0, DmccSave, true, false, false, false, false);
daq9513GetHold(DaqHandle00, DiodtLocal9513, 0, 1, count1);
while loop
  do
    begin
      daq9513MultCtrl(DaqHandle00, DiodtLocal9513, 0, DmccSave, true, false, false, false, false);
      daq9513GetHold(DaqHandle00, DiodtLocal9513, 0, 1, count2);
      StrainGauge1Voltage := Scandata.ReadOneChannel(Gauge1Chan);
      Series1.AddXY( (count2-count1), StrainGauge1Voltage);
      Series2.AddXY( (count2-count1), StrainGauge2Voltage);
      Series3.AddXY( (count2-count1), StrainGauge3Voltage);
          if count2 - count1>100 then Loop := false;
  end;
end;
end;
end;
unit GasGunInterface;

interface

uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs, Daqx, GasGunC;

Const
MaxChannels  = 16;
ADVOLTS      = 5;

var
DaqPort      : word;
err          : DaqError;

procedure CreateScanDataObject(StartChan,EndChan:DWORD; pts:longint;
Freq:Single; Ga : byte);

type
PScanData   = ^TScanData;
TScanData   = array [0..1] of Word;
PScanTags   = ^TScanTags;
TScanTags   = array [0..1] of Byte;
PRealData   = ^TRealData;
TRealData   = array [1..MaxChannels,0..1] of Single;

PReadScanObject = ^TReadScanObject;
TReadScanObject = object
  NoOfChannels : Integer;
  StartChannel, EndChannel : DWORD;
  PointsToAcquire : longint;
  Frequency : single;
  Gain : byte;
  OneShot : byte;
  Trigger : word;
  Level : byte;
  CalFactor : array [1..MaxChannels] of Single;
constructor Init(StartChan,EndChan:DWORD; pts:longint;
Freq:Single; Ga : byte);
destructor Done;
function ReadOneChannel(ChNo: integer): real;

protected
  TotalData : longint;

end;

var
ScanData   : PReadScanObject;
RawData    : PScanData;
Buffer     : PScanData;
Tags       : PScanTags;
RealData   : PRealData;

type
TGasGunInterface_Frm = class(TForm)
private
  { Private declarations }
public
  { Public declarations }
procedure InitRelays;
procedure TurnRelayOn(ChNo : integer);
procedure TurnRelayOff(ChNo : integer);

var
GasGunInterface_Frm: TGasGunInterface_Frm;
implementation

{$R *.DFM}

procedure TGasGunInterface_Frm.InitRelays;
var
    cfg : DWord;
    err : DaqError;
begin
    if daqIOGet8255Conf(DagHandle00, DIODevice, DIODevice, DIODevice, DIODevice, DIODevice, cfg) <> DerrNoError
    then showmessage('DIO Port ERROR');
    DIOConfig := cfg;
    DIODevice := DiodtLocal8255;
    DIODevice := Diodp8255A;
    DIOExpansionPort := DioepP1;
    DIOWhichDevice := 0;
end;

procedure TGasGunInterface_Frm.TurnRelayOn(ChNo : integer);
var
    error : DaqError;
    bitValue : longbool;
    cfg : DWord;
begin
    bitValue := true;  {Ord(true)=1}
    error := daqIOWriteBit(DagHandle00, DIODevice, DIODevice, DIODevice, DIODevice, ChNo, bitValue);
end;

procedure TGasGunInterface_Frm.TurnRelayOff(ChNo : integer);
var
    error : DaqError;
    bitValue : longbool;
    cfg : DWord;
begin
    bitValue := false;  {Ord(false)=0}
    error := daqIOWriteBit(DagHandle00, DIODevice, DIODevice, DIODevice, DIODevice, ChNo, bitValue);
end;

procedure CreateScanDataObject(StartChan, EndChan: DWORD; pts: longint; Freq: Single; Ga: byte);
begin
    New(ScanData, Init(StartChan, EndChan, pts, Freq, Ga));
end;

constructor TReadScanObject.Init(StartChan, EndChan: DWORD; pts: longint; Freq: Single; Ga: byte);
begin
    StartChannel := StartChan;
    EndChannel := EndChan;
    NoOfChannels := EndChannel - StartChannel + 1;
    PointsToAcquire := pts;
    Frequency := Freq;
    Gain := Ga;
    OneShot := 0;
    Level := 0;
    Trigger := 0;
    TotalData := NoOfChannels * PointsToAcquire;
    GetMem(RawData, TotalData * SizeOf(Word));
    GetMem(Buffer, TotalData * SizeOf(Word));
    GetMem(Tags, TotalData * SizeOf(Byte));
    GetMem(RealData, TotalData * SizeOf(Single) * MaxChannels);
end;
143: destructor TReadScanObject.Done;
144: begin
145: end;
146:
147: function TReadScanObject.ReadOneChannel(ChNo: integer): real;
148: var
149: Flags : integer;
150: Gains : integer;
151: sample : word;
152: Temp1 : real;
153: begin
154: Gains := DgainX1;
155: Flags := DafAnalog + DafSingleEnded + DafBipolar;
156: daqAdcRd(DaqHandle00, ChNo, sample, Gains, flags);
157: { sample := sample shr 0; } 
158: Temp1 := (sample-2048*16)*(ADVOLTS)/(2048*16);
159: Result := Temp1;
160: end;
161:
162: end.
unit Errex;

interface

uses DaqX, GasGunInterface, GasGunC;

const
DV_DaqBook00 = 0;

procedure TestError(Device: integer);

procedure SetTheErrorHandler;

procedure OpenTheDaq(Device: integer);

// Daqx error handler prototypes
procedure ErrorHandler(errCode: DaqError); stdcall;

implementation

uses SysUtils, Dialogs;

//********************************************************************
//  This unit demonstrates how to initialize a daqx device an
//    how to select and error handle
/
// Functions used:
//   daqOpen( daqName )
//   daqOnline( handle, online )
//   daqSetDefaultErrorHandler( errHandler )
//   daqSetErrorHandler( handle, errHandler )
//   daqAdcSetTrig( handle, triggerSource, rising, level, hysteresis, channel )
//   daqClose( handle )
//********************************************************************

procedure TestError(Device: integer);

var
    online: longbool;
begin
    // Error handler for daq errors.
    // Procedure to set the error handler. Complains and stops if unable.
    procedure ErrorHandler(errCode: DaqError); stdcall;
    var
        msg: string;
    begin
        if ord(errCode)=39 then
            begin
                ShowMessage('ERROR - Thermocouple Malfunction');
            end
        else
            begin
                msg:= 'Message from procedure ErrorHandler' + Chr(13) + Chr(10);
                msg:= msg + 'DaqComp error number ' + IntToStr(ord(errCode)) + ' occurred.';
                MessageDlg(msg , mtError, [mbOk], 0);
            end;
    end;
end;

procedure SetTheErrorHandler;
begin
(* if (daqSetErrHandler(addr(ErrorHandler)) <> DerrNoError ) then
begin
MessageDlg('Unable to set default error handler. Exiting program', mtError, [mbOk], 0);
Halt;
end;
HeatChamberMainFrm.Memo1.Lines.Add( 'Error handler set to procedure "ErrorHandler"');
// Disable the choice for the driver built-in error handler since it isn't available
// after setting another handler until the DAQX.DLL is reloaded.
HeatChamberMainFrm.DefaultErrBtn.Enabled:= False;
*)
end;

//********************************************************************************
procedure OpenTheDaq(Device: integer);
var
err: DaqError;
begin
DaqHandle00 := daqOpen('DaqBook0');
end;